AD-A104 165   UNIVERSITY OF CENTRAL FLORIDA ORLANDO DEPT OF MATHEM--ETC   F/G 12/1

A FORTRAN PROGRAM FOR ESTIMATING PARAMETERS IN A CUMULATIVE DIS--ETC(U)

MAR 81   S J BEAN, M HEUSER, P N SOMERVILLE   F19628-80-C-0004

UNCLASSIFIED   SCIENTIFIC-3   AFGL-TR-81-0120   NL

1 OF 1
AD
A104165

END
DATE
FILMED
10-81
DTIC

LEVEL II 12

AFGL-TR-81-0120

A FORTRAN PROGRAM FOR ESTIMATING PARAMETERS IN A CUMULATIVE DISTRIBUTION FUNCTION

Steven J. Bean
Mark Heuser
Paul N. Somerville

Scientific Report No. 3

31 March 1981

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICAL LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFGL-TR-81-0120	2. GOVT ACCESSION NO. AD-A104 165	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A FORTRAN PROGRAM FOR ESTIMATING PARAMETERS IN A CUMULATIVE DISTRIBUTION FUNCTION		5. TYPE OF REPORT & PERIOD COVERED Scientific ~~Report No.~~ - 3
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) S. J. Bean M. Heuser R. N. Somerville		8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0004
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Central Florida Department of Mathematics and Statistics P.O. Box 25000 Orlando, FL. 32816		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 667009 AF
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory Hanscom AFB, 01731 Contract Monitor/ I. Gringorten/LYD		12. REPORT DATE 31 March 1981
		13. NUMBER OF PAGES 17
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Non-linear Regression	Curve Fitting
FORTRAN Programming	Climatology
Visibility	Empirical Cumulative Distribution
WEIBULL	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report documents, and gives a listing for a FORTRAN program written to estimate the parameters of a cumulative distribution function which best fits an empirical cumulative distribution function in a least squares sense. Non-linear regression techniques are used. The program as listed uses the Weibull distribution for the fit, but with the replacment of certain modules the program may be used to fit many different distributions.

TABLE OF CONTENTS

1. Introduction

Given a large set of measurements of some quantity or variable, it is often useful to model the data using some statistical distribution function. For example if one has records of the "visibility" at Mildenhall, England for 10 a.m. February over a number of years, one may fit a Weibull distribution to the data. The Weibull distribution has two Parameters, and the values selected for the two parameters are the ones for which the model best fits the data.

This report documents a FORTRAN program that has been written to estimate the parameters of a statistical distribution function which best fits a set of measurements on some variable. The fit is "best" in the sense that the model cumulative distribution function and the empirical cumulative distribution function (from the data) are closest to each other in the least squares sense.

Suppose the measurements are ordered from smallest to largest, that is $x_{(1)} \le x_{(2)} \le x_{(3)} \le \ldots \le x_{(N)}$ where $x_{(i)}$ represents the i^{th} smallest measurement. Then the empirical cumulative distribution function may be defined as

$$\hat{F}_N(x) = \frac{2i-1}{2N} \quad \text{for } x_{(i)} \le x < x_{(i+1)}$$

$$= 0 \qquad \text{for} \qquad x < x_{(1)} \quad . \tag{1.1}$$

If $F(x;\theta)$ is the model cumulative distribution function, then the values for θ (θ may be a vector of values) which are chosen are those which minimize the expression

$$\sum_{i=1}^{N} \left[(2i-1)/(2N)-F(x_i;\theta)\right]^2 \quad . \tag{1.2}$$

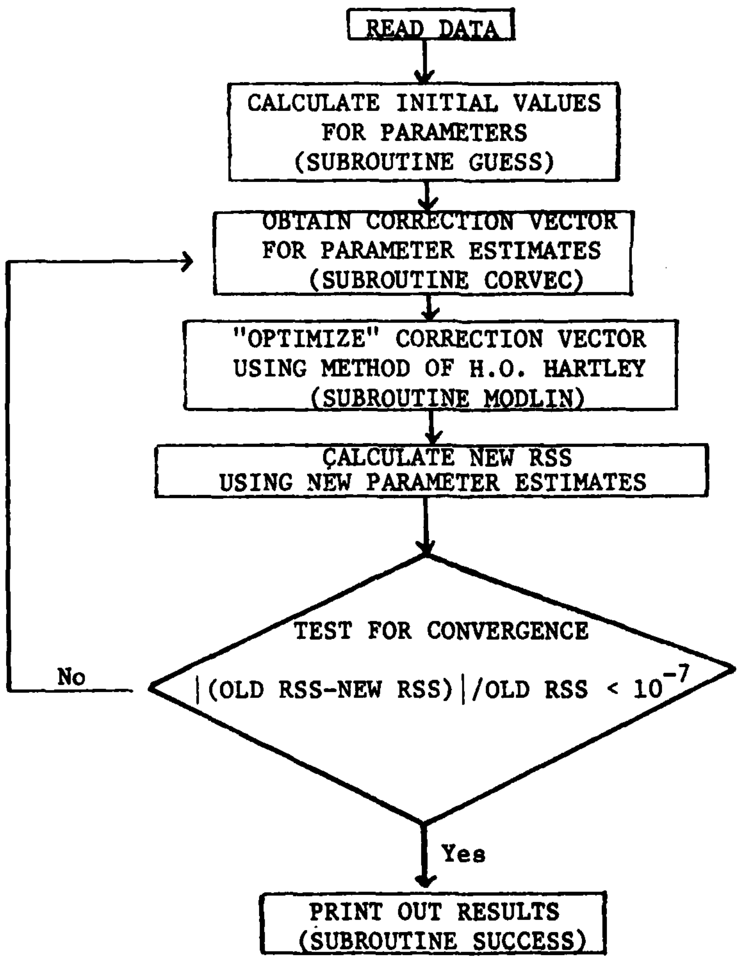
In the FORTRAN program, the determination of θ, is accomplished by non-linear regression techniques where $\hat{F}_N(x) = (2i-1)/(2N)$ for $i = 1,2,\ldots,N$, is the dependent variable. Expression (1.2), the quantity to be minimized, is the "Residual Sum of Squares". A detailed description of the techniques used to fit distributions to data using non-linear regression techniques is given in Heuser, Somerville and Bean (1980).

2. Flow Chart

In non-linear regression, the model may be written

$$y = F(x;\theta) + \epsilon \quad , \tag{2.1}$$

where θ represents a vector of unknown parameters.
The usual technique is to linearize $F(x;\theta)$ by the use of a first order Taylor Series expansion about guessed values θ_o. The expression (2.1) is then linear in $(\theta-\theta_o)$, and the usual least squares regression methods may be used to estimate $\theta-\theta_o$, the "correction" to the original guessed value. The procedure is then repeated with a first order Taylor Series expansion about the "corrected" guessed value for θ, the process terminating when the percentage reduction in residual sum of squares is less than some specified amounts. The flow chart below outlines the program.



3. **FORTRAN Code**

```
C***********************************************************************
C                                                                      *
C TITLE: WEIBULL NONLINEAR REGRESSION PROGRAM                          *
C        THE FOLLOWING PROGRAM REGRESSES VISIBILITY DATA ON THE        *
C        WEIBULL DISTRIBUTION.                                         *
C                                                                      *
C INPUT: N, X(1:N), ACC, AND Y(1:N) (SEE VARIABLE DICTIONARY)          *
C        N, X(1:N), AND ACC ARE INPUT ONCE IN THE BEGINNING OF THE     *
C        PROGRAM. Y(1:N) IS INPUT ONCE FOR EACH REGRESSION.            *
C                                                                      *
C OUTPUT: SUMMARY STATISTICS OF EACH REGRESSION INCLUDING ALPHA, BETA,*
C         MONTH, HOUR, SID, RMS, COUNT, X(1:N), Y(1:N), PREDICTED      *
C         VALUES OF THE DISTRIBUTION, AND THE RESIDUALS.               *
C                                                                      *
C SUBROUTINES: GUESS, SSE, WEIBULL, PSSEA, PSSEB, SUCCESS, FAIL,       *
C              SECANT, CORVEC, MODLIN                                  *
C                                                                      *
C FOR A GENERAL OVERVIEW OF THE REGRESSION PROBLEM, SEE 'LEAST         *
C SQUARES FITTING OF DISTRIBUTIONS USING NON-LINEAR REGRESSION' BY     *
C MARK HEUSER, PAUL SOMERVILLE, AND STEVE BEAN.                        *
C                                                                      *
C***********************************************************************
C
C***********************************************************************
C                                                                      *
C                          VARIABLE DICTIONARY                         *
C                                                                      *
C N:  THE NUMBER OF OBSERVATIONS (MAXIMUM OF 15)                       *
C X(1:N):  THE VALUES OF THE INDEPENDENT VARIABLE (DISTANCE)           *
C Y(1:N):  THE OBSERVED VISIBILITY PROBABILITIES AT EACH X             *
C ALPHA,BETA:  THE PARAMETERS IN THE WEIBULL MODEL                     *
C STARTA,STARTB:  THE STARTING VALUES FOR ALPHA AND BETA COMPUTED BY   *
C                 THE SUBROUTINE 'GUESS'                               *
C CORA,CORB:  THE CORRECTION VECTORS FOR ALPHA AND BETA COMPUTED BY    *
C             THE SUBROUTINE 'CORVEC'                                  *
C NRSS,ORSS:  THE RSS FOR TWO CONSECUTIVE ESTIMATES OF ALPHA AND       *
C             BETA.  NRSS IS FROM THE NEWER ESTIMATE; ORSS IS FROM     *
C             THE OLDER ESTIMATE.                                      *
C MONTH,HOUR,SID:  MONTH, HOUR, AND STATION IDENTIFIERS                *
C COUNT:  A LOOP COUNTER                                               *
C CONVERGE: A LOGICAL VARIABLE THAT INDICATES CONVERGENCE.             *
C ACC:  AN INTEGER VALUE CONTROLLING THE ACCURACY OF THE STARTING      *
C       VALUES FOR ALPHA AND BETA.  SEE SUBROUTINE 'GUESS'.            *
C                                                                      *
C***********************************************************************
C
```

```
      REAL ALPHA,BETA,X(15),Y(15),STARTA,STARTB,NRSS,ORSS,CORA,CORB
      REAL SSE,WEIBUL,RMS
      INTEGER SID,MONTH,HOUR,N,COUNT,ACC
      LOGICAL CONVERGE
      COMMON N,X,Y
C
      WRITE(6,200)             ! PRINT A TITLE
C
      READ,N                   ! INPUT THE NUMBER OF OBSERVATIONS
      READ,(X(I),I=1,N)        ! INPUT VALUES OF THE INDEPENDENT VARIABLE
      READ,ACC                 ! INPUT LEVEL OF ACCURACY OF STARTING VALUES
C
      ! THE FOLLOWING LOOP INPUTS AND REGRESSES ON THE EMPIRICAL
      ! DISTRIBUTION.  THE LOOP (AND THE PROGRAM) TERMINATES ON
      ! END OF FILE.
C
10    READ(5,100,END=40) (Y(I),I=1,N),SID,MONTH,HOUR
C
        CALL GUESS(STARTA,STARTB,ACC) ! GET STARTING VALUES FOR ALPHA
        ALPHA=STARTA                  ! AND BETA
        BETA=STARTB
        NRSS=SSE(ALPHA,BETA)
C
      ! THE FOLLOWING LOOP SOLVES FOR ALPHA AND BETA.  CONVERGENCE IS
      ! ASSUMED WHEN THE PROPORTIONAL CHANGE IN THE RSS FOR TWO CON
      ! SECUTIVE ESTIMATES IS LESS THAN 1E-7.
C
        COUNT=0                       ! INITIALIZE THE LOOP CONTROL
        CONVERGE=.FALSE.              ! VARIABLES
C
20      IF ((COUNT.GT.50).OR.(CONVERGE)) GOTO 30
          ORSS=NRSS
          CALL CORVEC(ALPHA,BETA,CORA,CORB)
          CALL MODLIN(ALPHA,BETA,CORA,CORB)
          NRSS=SSE(ALPHA,BETA)
          CONVERGE=ABS(ORSS-NRSS).LT.(ORSS*1.0E-7)
          COUNT=COUNT+1
        GOTO 20
C
30      IF (CONVERGE) THEN
          CALL SUCCESS(SID,MONTH,HOUR,ALPHA,BETA,NRSS,COUNT)
        ELSE
          CALL FAIL(SID,MONTH,HOUR,STARTA,STARTB,ALPHA,BETA,NRSS)
        END IF
C
      GOTO 10
C
40    STOP
C
100   FORMAT(1X,14F4.3,I5,I2,I1)
200   FORMAT(///,35X,'NONLINEAR REGRESSION OF THE WEIBULL MODEL ON ',
     $        'VISIBILITY DATA',///)
      END
```

```
C
C*********************************************************************
C                                                                    *
C WEIBUL IS A REAL FUNCTION THAT COMPUTES THE VALUE OF THE WEIBULL   *
C DISTRIBUTION FOR THE INPUT PARAMETERS X, ALPHA, AND BETA.  ALL     *
C COMMUNICATION WITH THE PROCEDURE IS THROUGH THE PARAMETER LIST     *
C AND FUNCTION NAME.                                                 *
C                                                                    *
C*********************************************************************
C
      REAL FUNCTION WEIBUL(X,ALPHA,BETA)
        REAL X,ALPHA,BETA
        WEIBUL=1.0-EXP(-ALPHA*(X**BETA))
        RETURN
        END
C
C*********************************************************************
C                                                                    *
C SSE IS A REAL FUNCTION THAT COMPUTES THE SUM OF THE SQUARED ERRORS *
C IN THE WEIBULL MODEL AS A FUNCTION OF ALPHA AND BETA.  COMMUNICA-  *
C TION WITH THE PROCEDURE IS DONE THROUGH THE PARAMETER LIST, THE    *
C FUNCTION NAME, AND THE COMMON BLOCK.                               *
C                                                                    *
C*********************************************************************
C
      REAL FUNCTION SSE(ALPHA,BETA)
        INTEGER I,N
        REAL ALPHA,BETA,X(15),Y(15),WEIBUL
        COMMON N,X,Y
        SSE=0.0
        DO 10 I=1,N
          SSE=SSE+(Y(I)-WEIBUL(X(I),ALPHA,BETA))**2
10      CONTINUE
        RETURN
        END
C
C*********************************************************************
C                                                                    *
C CORVEC IS A SUBROUTINE THAT COMPUTES THE CORRECTION VECTORS CORA   *
C AND CORB.  COMMUNICATION WITH THE PROCEDURE IS DONE THROUGH THE    *
C PARAMETER LIST AND THE COMMON BLOCK.  THE INPUT PARAMETERS ARE     *
C ALPHA AND BETA; OUTPUT PARAMETERS ARE CORA AND CORB.               *
C                                                                    *
C*********************************************************************
C
      SUBROUTINE CORVEC(ALPHA,BETA,CORA,CORB)
        INTEGER I,N
        REAL ALPHA,BETA,CORA,CORB
        REAL DERA,DERB,TEMP,RS,WEIBUL,C11,C12,C22,D1,D2
        REAL X(15),Y(15)
        COMMON N,X,Y
```

```
C
      C11=0.0           ! C11,C12,C22,D1, AND D2 REPRESENT A SYMMETRIC
      C12=0.0           ! SYSTEM OF 2 EQUATIONS IN 2 UNKNOWNS.  THE 2
      C22=0.0           ! UNKNOWNS ARE CORA AND CORB.  HERE C11,C12,C22,
      D1=0.0            ! D1, AND D2 ARE INITIALIZED TO 0.  IN THE DO
      D2=0.0            ! LOOP THAT FOLLOWS, THEIR VALUES ARE COMPUTED.
C
      DO 10 I=1,N
        RS=Y(I)-WEIBUL(X(I),ALPHA,BETA)     ! RS=OBS-EXP
        TEMP=X(I)**BETA
        DERA=TEMP*EXP(-ALPHA*TEMP)          ! DERIVATIVE WITH RESPECT
                                            ! TO ALPHA
        DERB=DERA*ALOG(X(I))*ALPHA          ! DERIVATIVE WITH RESPECT
                                            ! TO BETA
        C11=C11+(DERA**2)                   ! COMPUTE C11
        C12=C12+(DERA*DERB)                 ! COMPUTE C12
        C22=C22+(DERB**2)                   ! COMPUTE C22
        D1=D1+(DERA*RS)                     ! COMPUTE D1
        D2=D2+(DERB*RS)                     ! COMPUTE D2
10    CONTINUE
C
      TEMP=(C11*C22)-(C12**2)               ! NOW THE SYSTEM IS SOLVED
                                            ! USING CRAMER'S RULE
                                            ! TEMP IS THE DETERMINANT
      CORA=((D1*C22)-(D2*C12))/TEMP
      CORB=((C11*D2)-(D1*C12))/TEMP
      RETURN
      END
C
C*********************************************************************
C                                                                    *
C MODLIN IS A SUBROUTINE THAT IMPLEMENTS THE MODIFICATION OF THE     *
C LINEARIZATION METHOD PROPOSED BY H.O. HARTLEY IN HIS PAPER 'THE    *
C MODIFIED GAUSS-NEWTON METHOD FOR THE FITTING OF NON-LINEAR REGRESS-*
C ION FUNCTIONS BY LEAST SQUARES.'  ALL COMMUNICATION WITH THE PROC- *
C DURE IS DONE THROUGH THE PARAMETER LIST: ALPHA,BETA,CORA,CORB.     *
C MODLIN OPTIMIZES THE CORRECTION VECTORS COMPUTED BY CORVEC AND THEN*
C ADDS THEM TO ALPHA AND BETA.  WHEN THE PROCEDURE RETURNS, ALPHA AND*
C BETA ARE THE NEW PARAMETER ESTIMATES.  THE VALUES OF CORA AND CORB *
C MAY HAVE BEEN CHANGED IN THE PROCEDURE.                            *
C                                                                    *
C*********************************************************************
C
     SUBROUTINE MODLIN(ALPHA,BETA,CORA,CORB)
      REAL ALPHA,BETA,CORA,CORB
      REAL Q0,Q1,Q2,V,SSE,DENOM
      REAL TEMP
C
      ! LET THETA=(ALPHA,BETA) BE THE CURRENT PARAMETER VALUES AND
      ! DELTA=(CORA,CORB) BE THE CORRECTION VECTOR.  MODLIN ESTIMATES
      ! THE VALUE OF V>=0 THAT MINIMIZES SSE(THETA+V*DELTA).  SSE IS
      ! COMPUTED AT THETA (Q0), THETA+.5*DELTA (Q1), AND THETA+DELTA
      ! (Q2).  V IS FOUND SO THAT THETA+V*DELTA IS THE VERTEX OF THE
      ! PARABOLA PASSING THROUGH Q0, Q1, AND Q2.
```

```
C
      Q0=SSE(ALPHA,BETA)
      Q1=SSE(ALPHA+.5*CORA,BETA+.5*CORB)
      Q2=SSE(ALPHA+CORA,BETA+CORB)
C
10    DENOM=4.0*(Q2+Q0-(2.0*Q1))
C
      ! IF DENOM IS CLOSE TO ZERO THEN WE CAN'T COMPUTE V WITHOUT
      ! PRODUCING A DIVIDE-BY-ZERO OR FLOATING-POINT-OVERFLOW
      ! ERROR.  IN THIS CASE, ADD THE CORRECTION VECTOR ASSOCIATED
      ! WITH THE MINIMUM OF Q1 AND Q2 TO ALPHA AND BETA.
C
      IF (ABS(DENOM).LT.1E-15) THEN
        IF (Q1.LT.Q2) THEN
          ALPHA=ALPHA+.5*CORA
          BETA=BETA+.5*CORB
        ELSE
          ALPHA=ALPHA+CORA
          BETA=BETA+CORB
        END IF
        RETURN
      END IF
C
      V=.5+((Q0-Q2)/DENOM)
      TEMP=SSE(ALPHA+V*CORA,BETA+V*CORB)
C
      ! IF V<0 OR TEMP>Q0 THEN THE COMPUTATION OF V IS REDONE WITH
      ! DELTA=.5*DELTA.
C
      IF ((V.LT.0.0).OR.(TEMP.GT.Q0)) THEN
        CORA=.5*CORA
        CORB=.5*CORB
        Q2=Q1
        Q1=SSE(ALPHA+.5*CORA,BETA+.5*CORB)
        GOTO 10
      END IF
C
      ALPHA=ALPHA+V*CORA
      BETA=BETA+V*CORB
      RETURN
      END
C
C**********************************************************************
C                                                                     *
C GUESS IS A SUBROUTINE THAT FINDS STARTING VALUES FOR ALPHA AND      *
C BETA.  COMMUNICATION WITH THE PROCEDURE IS THROUGH THE PARAMETER    *
C LIST AND THE COMMON BLOCK.  ALPHA AND BETA ARE BOTH OUTPUT PARA-    *
C METERS; ACC IS AN INPUT PARAMETER.                                  *
C                                                                     *
C**********************************************************************
C
```

```
      SUBROUTINE GUESS(ALPHA,BETA,ACC)
        EXTERNAL PSSEA,PSSEB
        INTEGER I,J,N,E(3),ACC
        REAL ALPHA,BETA,C(3),D(3),T1,T2,PSSEA,PSSEB
        REAL X(15),Y(15)
        LOGICAL CONVERGE
        COMMON N,X,Y
C
        ! GIVEN TWO DATA POINTS IN THE EMPIRICAL DISTRIBUTION, WE CAN
        ! SOLVE FOR ALPHA AND BETA SO THAT THE WEIBULL MODEL FITS
        ! THROUGH THOSE TWO POINTS EXACTLY.  GUESS CHOOSES THREE DATA
        ! POINTS AND FITS THE WEIBULL THROUGH THE FIRST TWO, THE LAST
        ! TWO, AND THE FIRST AND LAST POINTS, THUS ARRIVING AT THREE
        ! DIFFERENT ESTIMATES FOR ALPHA AND BETA.  THE AVERAGES OF THE
        ! THREE ESTIMATES ARE USED AS STARTING VALUES FOR ALPHA AND
        ! BETA
C
        ALPHA=0.0
        BETA=0.0
        IF (Y(N).EQ.0.0) RETURN
C
        E(1)=2           ! THE VALUES OF E(1), E(2), AND E(3) DETERMINE
        E(2)=8           ! WHICH THREE DATA POINTS ARE CHOSEN.  HERE THE
        E(3)=13          ! 2ND, 8TH, AND 13TH POINTS ARE USED.
C
        DO 20 I=1,3
          D(I)=X(E(I))
          IF (Y(E(I)).EQ.0.0) THEN
            C(I)=.00001
          ELSE
            C(I)=Y(E(I))
          END IF
20      CONTINUE
C
        DO 30 I=1,3
          J=MOD(I,3)+1
          T1=ALOG(ALOG(1.0-C(I))/ALOG(1.0-C(J)))/ALOG(D(I)/D(J))
          BETA=BETA+T1
          ALPHA=ALPHA+(-ALOG(1.0-C(I))/D(I)**T1)
30      CONTINUE
C
        ALPHA=ALPHA/3.0
        BETA=BETA/3.0
C
        ! MOST OF THE TIME THESE STARTING VALUES WILL BE GOOD ENOUGH TO
        ! BEGIN THE NONLINEAR REGRESSION PROCEDURE.  SOME CASES,
        ! HOWEVER, WILL REQUIRE EVEN MORE ACCURATE STARTING VALUES.
        ! THE FOLLOWING CODE OPTIONALLY IMPROVES THE STARTING VALUES,
        ! DEPENDING ON THE VALUE OF ACC.  ACC IS MERELY THE
        ! NUMBER OF TIMES THE LOOP BELOW IS EXECUTED.  THE LOOP TRIES
        ! TO OPTIMIZE ALPHA FOR A FIXED BETA, AND THEN OPTIMIZES BETA
        ! FOR A FIXED ALPHA.
```

```
C
         IF (ACC.LE.0) RETURN
C
         DO 40 I=1,ACC
           T1=ALPHA
           T2=BETA
           CALL SECANT(ALPHA,BETA,ALPHA,PSSEA,CONVERGE)
           IF (CONVERGE) CALL SECANT(ALPHA,BETA,BETA,PSSEB,CONVERGE)
           IF (.NOT.(CONVERGE)) GOTO 50
40       CONTINUE
         RETURN
50       ALPHA=T1
         BETA=T2
         RETURN
         END
C
C*********************************************************************
C                                                                    *
C FAIL IS AN OUTPUT ROUTINE CALLED WHEN A DISTRIBUTION FAILS TO CON- *
C VERGE AFTER 50 ITERATIONS.  THE VALUES OF SEVERAL VARIABLES ARE    *
C WRITTEN TO THE OUTPUT FILE.  COMMUNICATION WITH THE PROCEDURE IS   *
C DONE THROUGH THE PARAMETER LIST AND THE COMMON BLOCK.  ALL PARA-   *
C METERS ARE INPUT PARAMETERS.                                       *
C                                                                    *
C*********************************************************************
C
       SUBROUTINE FAIL(SID,MONTH,HOUR,STARTA,STARTB,ALPHA,BETA,NRSS)
         INTEGER SID,MONTH,HOUR,N,I
         REAL STARTA,STARTB,ALPHA,BETA,NRSS,TEMP,X(15),Y(15),SSE
         COMMON N,X,Y
C
         TEMP=SSE(STARTA,STARTB)
         WRITE(6,100)
         WRITE(6,200) SID,MONTH,HOUR
         WRITE(6,300)
         WRITE(6,400) STARTA,STARTB,TEMP
         WRITE(6,500) ALPHA,BETA,NRSS
         WRITE(6,600)
         WRITE(6,700)
         DO 10 I=1,N
           WRITE(6,800) X(I),Y(I)
10       CONTINUE
         RETURN
C
100    FORMAT(///,T32,35('* '),///)
200    FORMAT(/,1X,'ATTENTION: STATION ',I2,', MONTH ',I2,', HOUR ',
      $        I1,' FAILED TO',/,1X,'CONVERGE AFTER 50 ITERATIONS.')
300    FORMAT(/,1X,'A VARIABLE DUMP FOLLOWS:',/)
400    FORMAT(1X,'STARTA=',G15.7,'  STARTB=',G15.7,
      $        '  SSE(STARTA,STARTB)=',G15.7)
500    FORMAT(1X,'ALPHA=',G15.7,'    BETA=',G15.7,
      $        '     SSE(ALPHA,BETA)=',G15.7)
600    FORMAT(/,5X,'ENDPTS',14X,'OBCUMFR')
700    FORMAT('+',4X,'______',14X,'_______',/)
800    FORMAT(1X,G15.7,5X,G15.7)
       END
```

```
C
C******************************************************************
C                                                                *
C SUCCESS IS AN OUTPUT ROUTINE CALLED WHEN A DISTRIBUTION CONVERGES *
C SUCCESSFULLY WITHIN 50 ITERATIONS.  SUMMARY STATISTICS OF THE   *
C REGRESSION ARE WRITTEN TO THE OUTPUT FILE.  COMMUNICATION WITH  *
C THE PROCEDURE IS THROUGH THE PARAMETER LIST AND COMMON  BLOCK.  *
C ALL PARAMETERS ARE INPUT PARAMETERS.                            *
C                                                                *
C******************************************************************
C
      SUBROUTINE SUCCESS(SID,MONTH,HOUR,ALPHA,BETA,NRSS,COUNT)
        INTEGER SID,MONTH,HOUR,I,N,COUNT
        REAL ALPHA,BETA,NRSS,RMS,X(15),Y(15),EX,RS,WEIBUL
        COMMON N,X,Y
C
        RMS=SQRT(NRSS/FLOAT(N))
        WRITE(6,100)
        WRITE(6,200)
        WRITE(6,300)
        WRITE(6,400) SID,MONTH,HOUR,ALPHA,BETA,RMS,COUNT
        WRITE(5,500)
        WRITE(6,600)
        DO 10 I=1,N
          EX=WEIBUL(X(I),ALPHA,BETA)
          RS=Y(I)-EX
          WRITE(6,700) X(I),Y(I),EX,RS
10      CONTINUE
        RETURN
C
100   FORMAT(///,T31,35('* '),///)
200   FORMAT(10X,'STATION ID',5X,'MONTH',5X,'HOUR',10X,'ALPHA',15X,
     $         'BETA',17X,'RMS',11X,'# OF ITERATIONS')
300   FORMAT('+',9X,'_______ __',5X,'_____',5X,'____',10X,'_____',15X,
     $         '____',17X,'___',11X,'_ __ __________',//)
400   FORMAT(12X,I6,2(8X,I2),6X,G14.7,4X,G14.7,4X,G14.7,11X,I3,///)
500   FORMAT(T38,'ENDPTS',5X,'OBCUMFR',10X,'EXCUMFR',12X,'RESIDUAL')
600   FORMAT('+',T38,'______',5X,'_______',10X,'_______',12X,'________',
     $         //)
700   FORMAT(T37,F7.4,6X,F5.3,7X,G15.7,5X,G15.7)
      END
C
C******************************************************************
C                                                                *
C SECANT IS A SUBROUTINE THAT USES THE SECANT METHOD OF ROOT SOLVING  *
C TO FIND THE ROOT OF PSSEA HOLDING BETA CONSTANT, OR TO FIND THE   *
C ROOT OF PSSEB HOLDING ALPHA CONSTANT.  (THIS MEANS IT WILL FIND THE *
C BEST ALPHA FOR A GIVEN BETA, OR THE BEST BETA FOR A GIVEN ALPHA.)   *
C COMMUNICATION WITH THE PROCEDURE OCCURS THROUGH THE PARAMETER LIST  *
C AND THE COMMON BLOCK.  IN THE PARAMETER LIST, ALPHA AND BETA ARE    *
C THE CURRENT VALUES OF THE MODEL PARAMETERS.  PARM IS THE VARIABLE   *
C BEING OPTIMIZED (EITHER ALPHA OR BETA), AND PDER IS THE PARTIAL     *
C DERIVATIVE FUNCTION (EITHER PSSEA OR PSSEB). CONVERGE IS A LOGICAL  *
C VARIABLE INDICATING WHETHER THE SECANT METHOD CONVERGED ON A ROOT.  *
C TO SOLVE FOR THE ROOT OF PSSEA, SET PARM=ALPHA AND PDER=PSSEA. THE  *
C OPTIMIZED VALUE OF ALPHA WILL BE RETURNED.  TO SOLVE FOR THE ROOT OF*
C PSSEB, SET PARM=BETA AND PDER=PSSEB.  THE OPTIMIZED VALUE OF BETA   *
C WILL BE RETURNED.                                                   *
C                                                                     *
C*********************************************************************
```

```
C
      SUBROUTINE SECANT(ALPHA,BETA,PARM,PDER,CONVERGE)
        REAL ALPHA,BETA,PARM,PDER,X(15),Y(15),T1,T2,T3,DELTA
        INTEGER N,I
        LOGICAL CONVERGE
        COMMON N,X,Y
C
        T2=PDER(ALPHA,BETA)
        DELTA=.001
        PARM=PARM+DELTA
        CONVERGE=.TRUE.
C
        DO 10 I=1,15
          T1=PDER(ALPHA,BETA)
          T3=T1-T2
          IF (ABS(T3).LE.1E-15) GOTO 15
          DELTA=(-T1*DELTA)/(T3)
          PARM=PARM+DELTA
          IF (ABS(DELTA).LE.1E-7) GOTO 20
          T2=T1
10      CONTINUE
C
15      CONVERGE=.FALSE.
20      RETURN
        END
C
C*********************************************************************
C                                                                    *
C PSSEA IS A REAL FUNCTION THAT COMPUTES THE PARTIAL DERIVATIVE OF SSE*
C WITH RESPECT TO ALPHA.  COMMUNICATION WITH THE PROCEDURE OCCURS    *
C THROUGH THE FUNCTION NAME, COMMON BLOCK, AND PARAMETER LIST.  ALPHA *
C AND BETA ARE INPUT PARAMETERS; THE VALUE OF THE DERIVATIVE IS      *
C RETURNED THROUGH THE FUNCTION NAME.                                *
C                                                                    *
C*********************************************************************
C
      REAL FUNCTION PSSEA(ALPHA,BETA)
        REAL ALPHA,BETA,X(15),Y(15),T1,T2
        INTEGER N,I
        COMMON N,X,Y
C
        PSSEA=0.0
        DO 10 I=1,N
          T1=X(I)**BETA
          T2=EXP(-ALPHA*T1)
          PSSEA=PSSEA+(Y(I)-1.0+T2)*(-T1*T2)
10      CONTINUE
C
        RETURN
        END
```

```
C***********************************************************************
C                                                                      *
C PSSEB IS A REAL FUNCTION THAT COMPUTES THE PARTIAL DERIVATIVE OF     *
C SSE WITH RESPECT TO BETA.  COMMUNICATION WITH THE PROCEDURE OCCURS   *
C THROUGH THE FUNCTION NAME, COMMON BLOCK, AND PARAMETER LIST.  ALPHA  *
C AND BETA ARE INPUT PARAMETERS; THE VALUE OF THE DERIVATIVE IS        *
C RETURNED THROUGH THE FUNCTION NAME.                                  *
C                                                                      *
C***********************************************************************
C
      REAL FUNCTION PSSEB(ALPHA,BETA)
        REAL ALPHA,BETA,X(15),Y(15),T1,T2,T3,T4
        INTEGER N,I
        COMMON N,X,Y
C
        PSSEB=0.0
        DO 10 I=1,N
          T1=X(I)**BETA
          T2=EXP(-ALPHA*T1)
          T3=-ALPHA*ALOG(X(I))*T1*T2
          T4=Y(I)-1.0+T2
          PSSEB=PSSEB+T3*T4
10      CONTINUE
C
        RETURN
        END
```

NONLINEAR REGRESSION OF THE WEIBULL MODEL ON VISIBILITY DATA

* *

STATION ID	MONTH	HOUR	ALPHA	BETA	RMS	# OF ITERATIONS
1	1	1	0.3001646	0.6776246	0.2175113E-01	5

ENDPTS	OBCUMFR	EXCUMFR	RESIDUAL
0.2500	0.110	0.1107035	-0.7034689E-03
0.3125	0.159	0.1275731	0.3142692E-01
0.5000	0.202	0.1711044	0.3089564E-01
0.6250	0.216	0.1961113	0.1988868E-01
0.7500	0.229	0.2188597	0.1014027E-01
1.0000	0.235	0.2593037	-0.2430369E-01
1.2500	0.276	0.2947218	-0.1872179E-01
1.5000	0.305	0.3263727	-0.2137274E-01
2.0000	0.355	0.3812875	-0.2628747E-01
2.5000	0.434	0.4279263	0.6073684E-02
3.0000	0.437	0.4684348	-0.3143480E-01
4.0000	0.553	0.5360345	0.1696545E-01
5.0000	0.618	0.5906984	0.2730155E-01
6.0000	0.642	0.6360627	0.5937338E-02

4. Some Remarks About the Program

The program given in this report was written to fit the Weibull distribution to visibility data. The visibility data was that contained in the "Revised Uniform Summary of Weather Observations" (RUSSWO's) prepared by the Data Processing Division of the Air Weather Service. Some changes must be made to fit another distribution to another variable. The program is made up of a series of subroutines and functions so that these may be altered to fit the users need without changing the flow of the program.

The function WEIBUL must be replaced by the desired cumulative distribution function (CDF). Also, the name of the function, WEIBUL, must be changed throughout the program if the name of the function is changed. The subroutine GUESS which gives the initial values of the parameters must be changed to correspond to the distribution being used. The same basic idea can be used. However, it may take more programming for other distributions particularly if the distribution is not in closed form. The subroutine CORVEC must be changed where the partial derivatives DERA and DERB occur. Another change is required in the functions PSSEA and PSSEB which are functions which take partials with respect to each of the parameters of the sum of squared errors. The appropriate partials must replace those of the Weibull distribution in each of these functions.

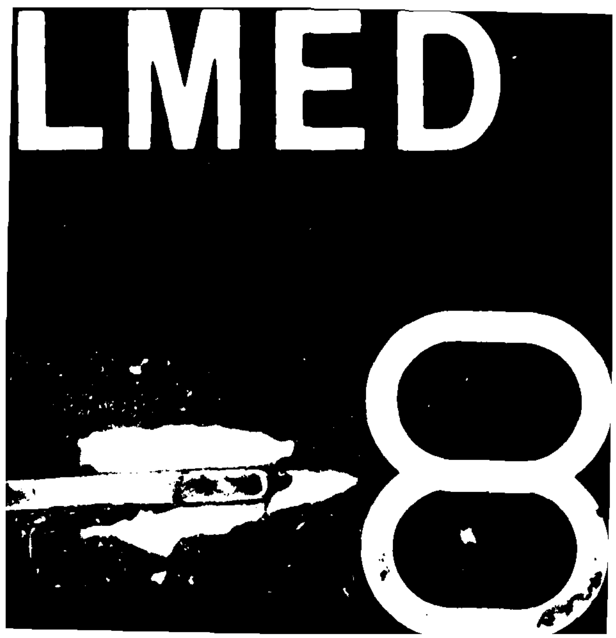
The program was designed to run in a batch environment, and the rules of standard FORTRAN - IV were adhered to as closely as possible. The only departures from FORTRAN - IV are the use of the IF-THEN-ELSE-ENDIF structure commonly found in FORTRAN 77, and the use of the exclamation mark to permit comments on the same line as code. Both deviations were made merely for clarity's sake in the program.

5. Sample Output

A sample output is shown in page 14.

6. Reference

Heuser, M. L., P. N. Somerville, and S. J. Bean "Least Squares Fitting of Distributions Using Non-linear Regression". AFGL-TR-80-0362.


LMED
8